Last time

- Finished introduction and overview:
  - Network access and physical media
  - Internet structure and ISPs
  - Delay & loss in packet-switched networks
  - Protocol layers, service models

This time

- Link layer
  - Overview
  - Error detection and correction
  - PPP

Chapter 5: The Data Link Layer

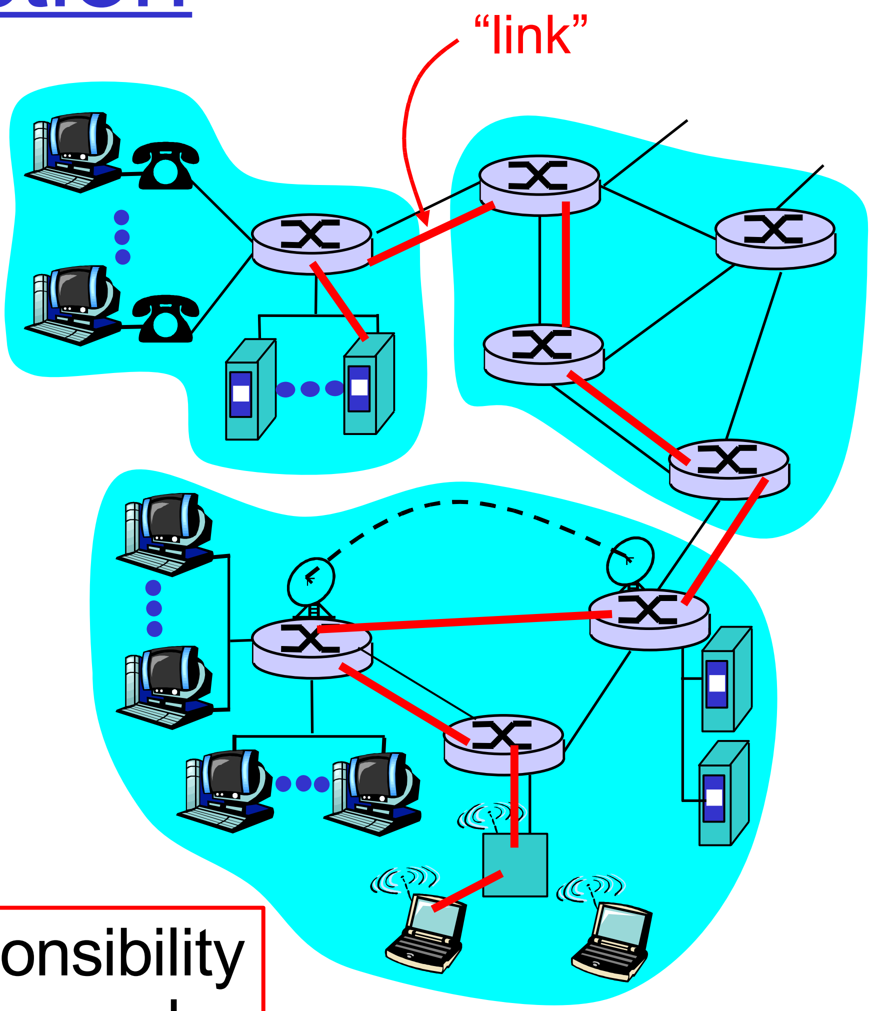
Our goals:

- Understand principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - reliable data transfer, flow control
- Instantiation and implementation of various link layer technologies

Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM and MPLS

Link Layer: Introduction



Some terminology:

- Hosts and routers are **nodes**
- Communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- A layer-2 packet is a **frame**, and encapsulates a layer-3 datagram

The data-link layer has the responsibility of transferring datagrams from one node to an adjacent node over a link

Link layer: context

- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Each link protocol provides different services
  - e.g., may or may not provide reliable delivery over link

Transportation analogy

- Trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- Tourist = datagram
- Transport segment = communication link
- Transportation mode = link layer protocol
- Travel agent = routing algorithm

Link Layer Services

Different link layers offer different combinations of these services:

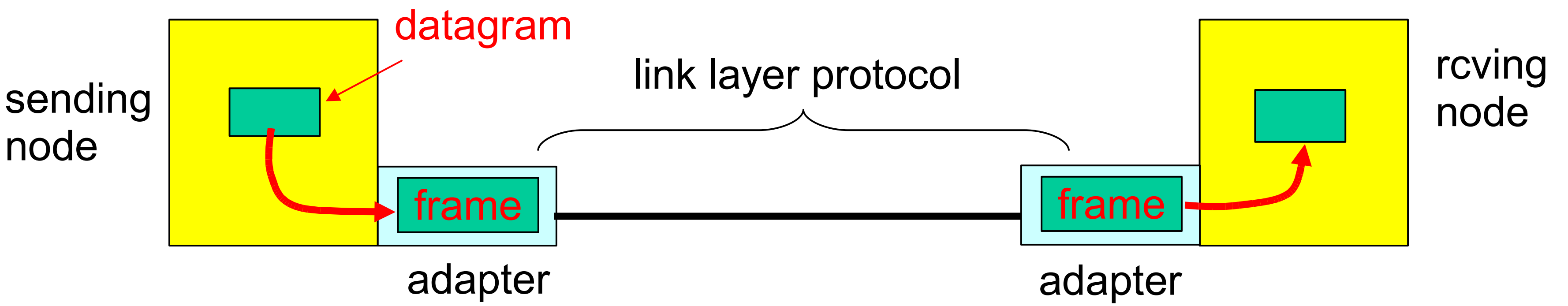
- Framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, destination

- Reliable delivery between adjacent nodes
  - based on acknowledgements and retransmissions
    - details later in the course (transport layer)
  - seldom used on low bit error link (fiber, some twisted pair)
  - wireless links: high error rates
    - Q: why both link-level and end-to-end reliability?

Link Layer Services (more)

- Flow Control:
  - pacing between adjacent sending and receiving nodes
- Error Detection:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- Error Correction:
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- Half-duplex and full-duplex:
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Adapters Communicating



- Link layer implemented in "adapter" (aka NIC)
  - Ethernet card, PCMCIA card, 802.11 card
- Sending side:
  - encapsulates datagram in a frame
  - adds error checking bits, flow control, etc.
  - medium access control
- Receiving side:
  - looks for errors, flow control, etc
  - extracts datagram, passes to receiving node
- Adapter is semi-autonomous
- Link & physical layers

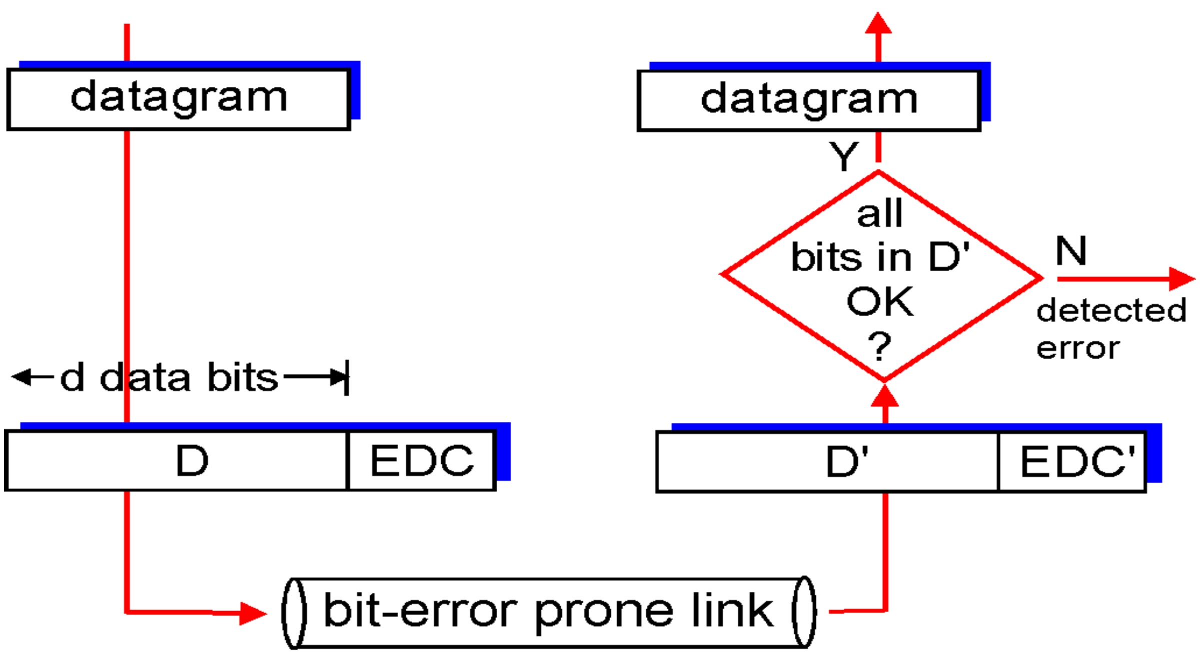
Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM

Error Detection

EDC= Error Detection and Correction bits (redundancy)
D = Data protected by error checking, may include header fields

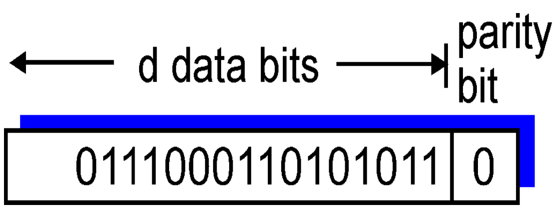
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity Checking

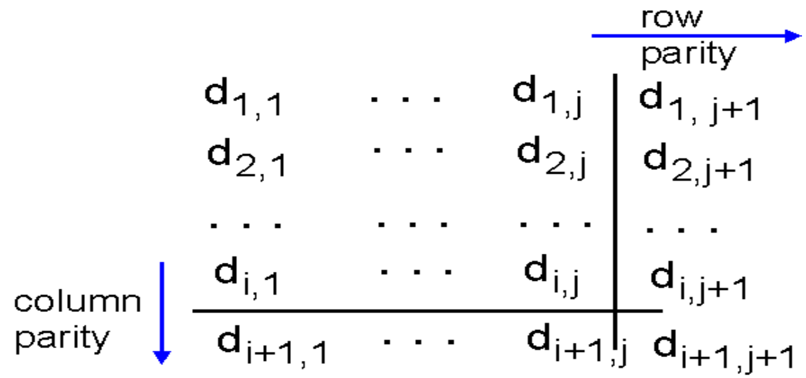
Single Bit Parity:

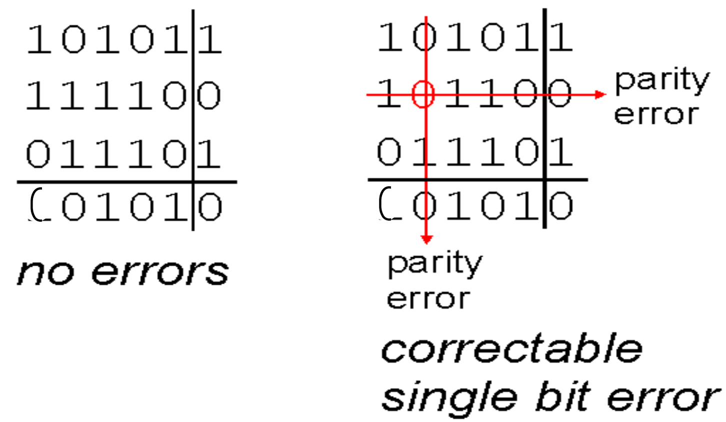
Detect single bit errors



Two Dimensional Bit Parity:

Detect *and correct* single bit errors





Internet checksum

Goal: detect "errors" (e.g., flipped bits) in transmitted segment; easy to implement in software

Sender:

- Treat segment contents as sequence of 16-bit integers in 1's complement notation
- Checksum:
  - add integers
  - invert sum
- Put checksum value into checksum field

Receiver:

- Compute checksum of received segment
- Check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?*

Internet Checksum Example

- Note
  - When adding numbers in 1’s complement notation, a carry-over from the most significant bit needs to be added to the result

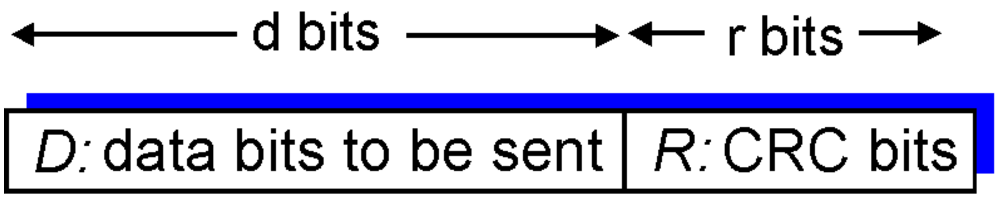
- Example: add two 16-bit integers

		1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
		1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
sum	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
																	1
add wraparound		1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum		0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

Checksumming: Cyclic Redundancy Check

- Sender has data D, which is d bits long.
- Sender computes R, an r-bit CRC of those d bits
  - r is usually 8, 12, 16, or 32
- Sender sends D followed by R to the receiver

- Receiver computes the CRC of the received data
  - If no match, there were certainly errors
  - If match, no error detected



Checksumming: Cyclic Redundancy Check

- Better at detecting errors than other methods we've seen
  - Can detect all burst errors up to r bits
  - But much more complicated to compute
- Since the link layer is handled semi-autonomously in adapter hardware, CRCs are more appropriate for the link layer than for higher layers in the protocol stack.
- The mathematical details are in the text.

Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM

Point to Point Data Link Control

- This is the simplest kind of link layer
  - One sender
  - One receiver
  - One link
- The most popular such protocol is PPP (point-to-point protocol)
  - You probably use this at home to connect to the Internet
    - Dialup: plain PPP
    - ADSL: PPP over Ethernet (PPPoE)

PPP Design Requirements [RFC 1557]

- packet framing: encapsulation of network-layer datagram in data link frame
  - carry network layer data of any network layer protocol (not just IP) *at the same time*
  - ability to demultiplex upwards
- bit transparency: must carry any bit pattern in the data field
- error detection (no correction)
- connection liveness: detect, signal link failure to network layer
- network layer address negotiation: endpoint can learn/configure each other's network address

PPP non-requirements

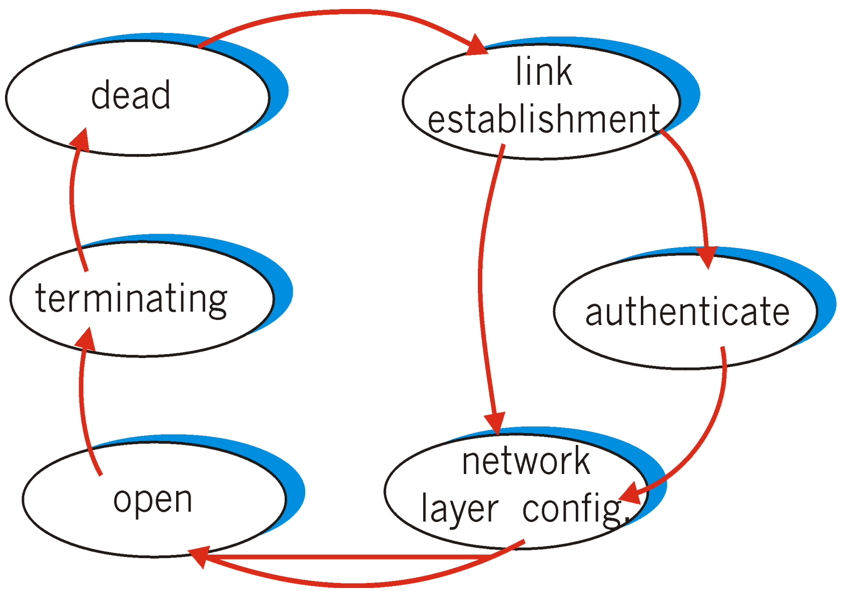
- no error correction/recovery
- no flow control
- out of order delivery OK
- no need to support multipoint links (e.g., polling)

Error recovery, flow control, data re-ordering
all relegated to higher layers!

PPP Link Control Protocol

Before exchanging network-layer data, data link peers must

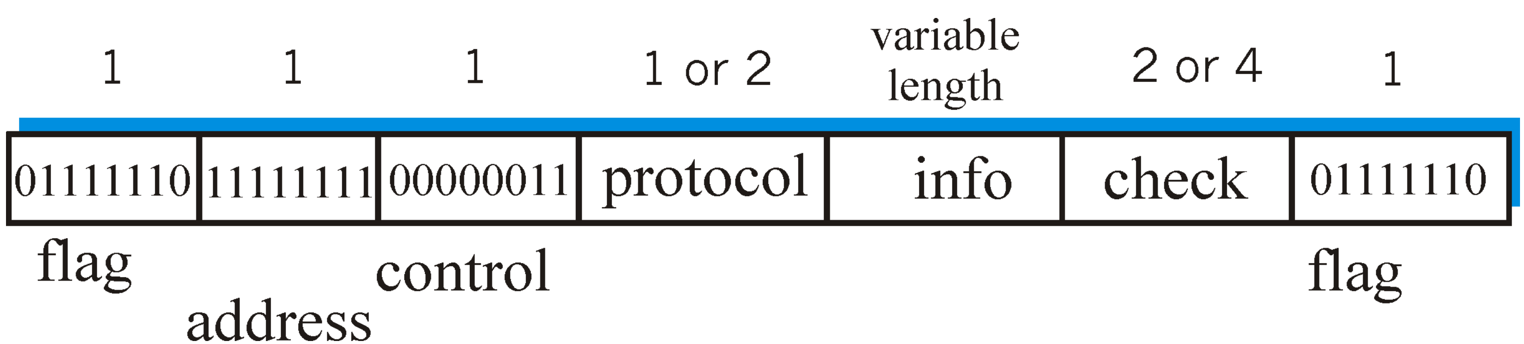
- configure PPP link (max. frame length, authentication)
- learn/configure network layer information



These are straightforward two-party protocols, because PPP is point-to-point

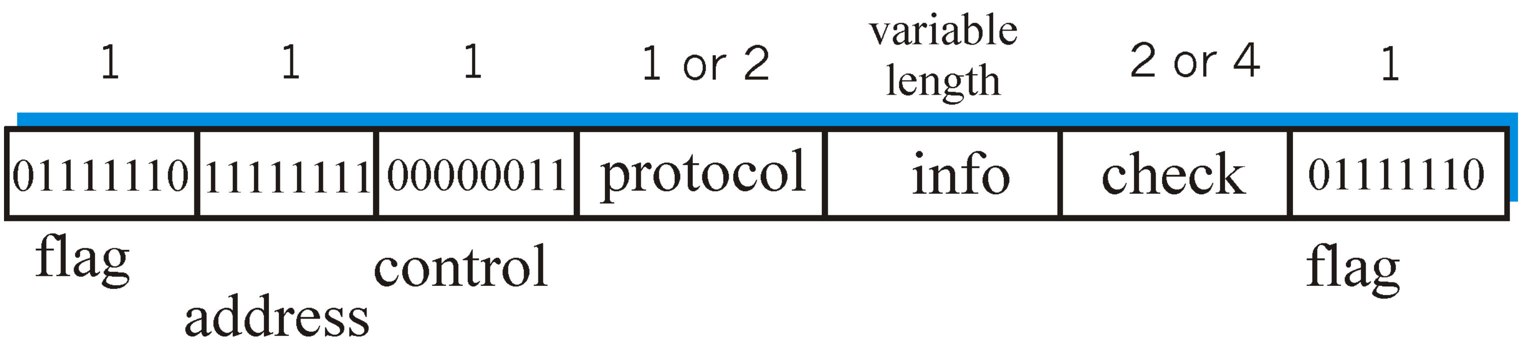
PPP Data Frame

- Flag: delimiter (framing)
- Address: does nothing (only one option)
- Control: does nothing; in the future possible multiple control fields
- Protocol: upper layer protocol to which frame delivered (e.g. PPP-LCP, IP, Appletalk, etc.)



PPP Data Frame

- Info: upper layer data being carried
- Check: cyclic redundancy check for error detection

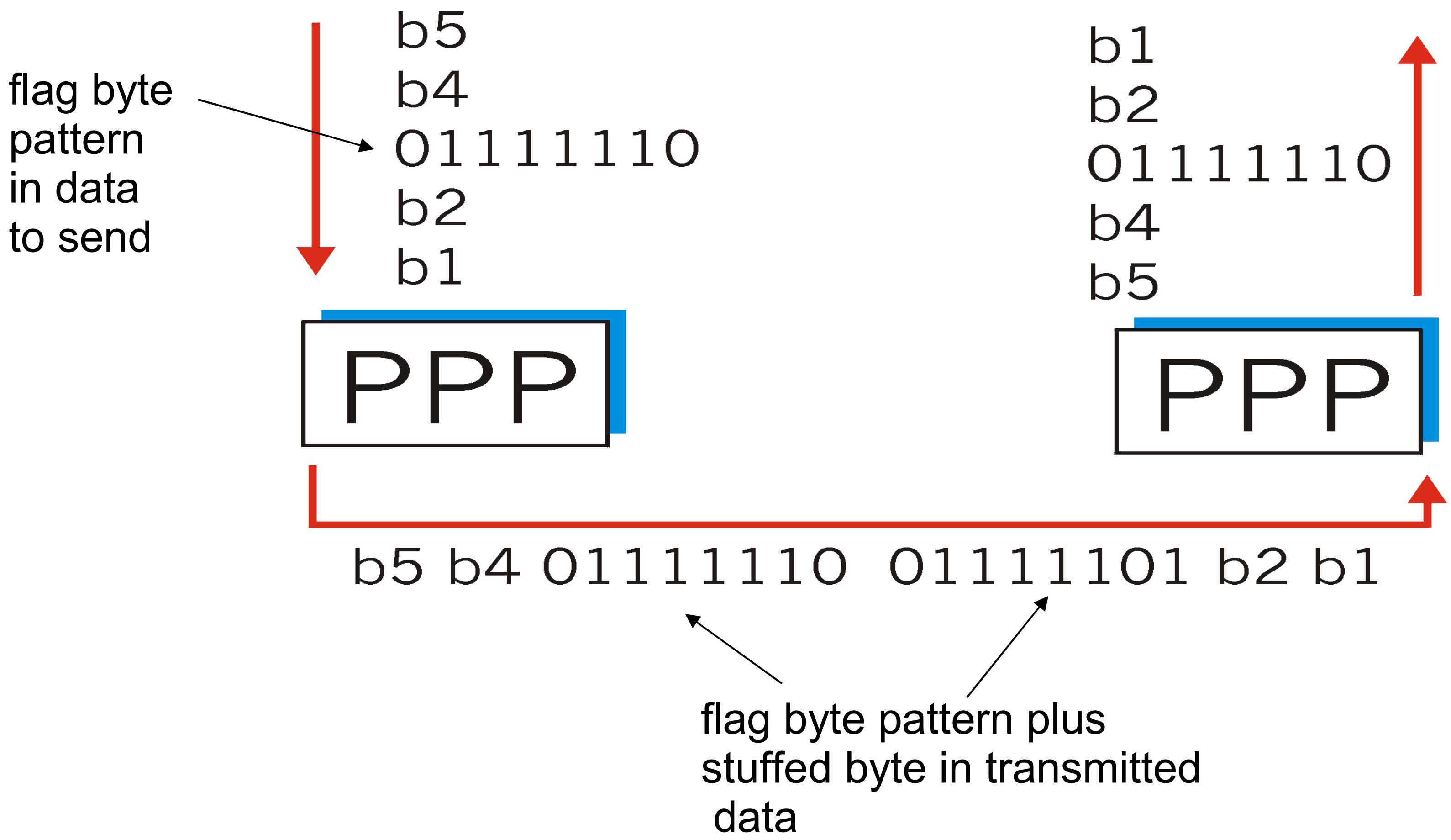


Byte Stuffing

- "Data transparency" requirement: protocol, info, checksum fields must be allowed to include flag pattern <01111110>
  - Q: is received <01111110> the closing flag, or part of some other field?

- Sender: adds ("stuffs") escape sequence <01111101> byte before:
  - each non-flag <01111110> byte
  - each <01111101> byte
- Receiver:
  - discard <01111101> byte and treat next as data
  - single <01111110>: flag byte

Byte Stuffing



Recap

- Link layer overview
  - Services
  - Adapters
- Error detection and correction
  - Parity check
  - Internet checksum
  - CRC
- PPP
  - Byte stuffing

Next time

- Multiple access protocols
- Link-layer addressing
- Ethernet